



THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: Foster D. Hinshaw, Raymond J. Andraka, David L. Meyers, Sharon L. Miller,
Michael Sporer, William K. Stewart and Barry M. Zane

Application No.: 10/667,203 Group: 2167

Filed: September 18, 2003 Examiner: Kimberly M. Lovel

Confirmation No.: 7168

For: INTELLIGENT STORAGE DEVICE CONTROLLER

CERTIFICATE OF MAILING OR TRANSMISSION	
I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as First Class Mail in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, or is being facsimile transmitted to the United States Patent and Trademark Office on:	
<u>10-1-08</u>	<u>Jane Morgan</u>
Date	Signature
<u>JANE MORGAN</u>	
Typed or printed name of person signing certificate	

APPEAL BRIEF

Mail Stop Appeal Brief Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Appeal Brief is submitted pursuant to the Notice of Appeal received in the U.S. Patent and Trademark Office (hereinafter, the Office) on July 7, 2008, and in support of the appeal from the final rejections set forth in the Final Office Action mailed from the Office on June 11, 2008. The fee for filing a brief in support of an appeal is enclosed. A Petition for Extension of Time and the appropriate fee are being filed concurrently.

TABLE OF CONTENTS

I.	REAL PARTY IN INTEREST.....	3
II.	RELATED APPEALS AND INTERFERENCES	3
III.	STATUS OF CLAIMS	3
IV.	STATUS OF AMENDMENTS.....	3
V.	SUMMARY OF CLAIMED SUBJECT MATTER.....	4
VI.	GROUND OF REJECTION TO BE REVIEWED ON APPEAL	6
VII.	ARGUMENT.....	7
1.	The Examiner Fails to Meet the Burden of Showing a Prima Facie Case of Obviousness.....	8
A.	Neither Baker nor Hausman performs primitive initial processing functions on data from a streaming data source; Baker and Hausman simply route data within the same data processor and to clients requesting it, respectively.....	9
B.	Housman neither selects fields to be assembled into output tuples nor assembles fields into tuples; Hausman merely passes complete records to clients requesting them	10
C.	Housman does not determine field boundaries in output data; Hausman reviews fields already delineated at the streaming data source	12
D.	Hausman neither asserts a use or lose decision nor prevents the output tuple from being transferred according to a use or lose decision value; Hausman simply routes or filters records	13
E.	Hausman does not contain logic; Hausman selects records based in software	14
2.	It would not be obvious to combine the teachings of Baker and Hausman.....	15
A.	There is no teaching, suggestion or motivation to so combine the references	16
B.	The proposed modification of Baker by Hausman would render Baker unsatisfactory for its intended purpose	16
C.	The proposed combination would not result in the claimed invention.....	17
3.	Claim Rejections.....	18
4.	Conclusion.....	22
VIII.	CLAIMS APPENDIX	23
IX.	EVIDENCE APPENDIX	26
X.	RELATED PROCEEDINGS APPENDIX.....	27

I. REAL PARTY IN INTEREST

The real party in interest is Netezza Corporation, 200 Crossing Boulevard, Framingham, Massachusetts 01702. Netezza Corporation is the Assignee of the entire right, title and interest in the subject application, by virtue of an Assignment recorded on March 15, 2004 at Reel 015072, Frames 0303-0309.

II. RELATED APPEALS AND INTERFERENCES

Applicants, the undersigned Attorney and Assignee are not aware of any presently pending related appeals or interferences which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal. Applicants do note, however, that a related application, U.S. Patent Application No. 10/665,726, filed September 18, 2003, has received a Final Office Action mailed from the Office on July 25, 2008. Applicants may file an appeal in that related application as well.

III. STATUS OF CLAIMS

Claims 1 through 14 have been finally rejected, and a copy appears in the Appendix of this Brief. Claims 1, 5 and 6 were amended in the Amendment filed on March 14, 2008. Claims 1-3, 7, 8, 10, 13 and 14 were amended in the Amendment filed on August 17, 2007. Claims 1, 9 and 11 were amended in the Amendment filed on August 4, 2006. Claims 4 and 12 appear as originally filed. Claims 1-14 are being appealed herein.

IV. STATUS OF AMENDMENTS

All prior amendments have been entered in the application. No amendments are being filed concurrently with this appeal brief or have been filed subsequent to an Advisory Action Before the Filing of an Appeal Brief dated July 28, 2008.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Claim 1

Claim 1 is directed to a Programmable Streaming Data Processor (PSDP) which is arranged to perform primitive functions directly on data received from a streaming data interface. The PSDP processes data from a streaming data source, such as a disk drive, prior to its being forwarded to a central processing unit (CPU) of a more general purpose processor called a Job Processing Unit (JPU). The PSDP performs certain preliminary processing in order to reduce the computational load on the local CPU.

The PSDP may have processing logic known as a Data Engine that is capable of examining fields of a record to determine whether a record will or will not be passed to the CPU of the JPU as an output tuple. An output tuple is comprised of the fields of the source record from the disk that are to be selected for further processing by the CPU and PSDP generated fields. As data streams out of the filter, an output tuple is formed in a First In, First Out (FIFO) memory, in a way that permits aborting the tuple if the filter logic determines that the particular tuple should not be passed on to the CPU.

Support for Claim 1 can be found at least in the Abstract; the specification at page 4, lines 1-4; page 5, lines 8-17; page 7; page 8, lines 1-3; as well as FIGS. 1 and 2, reference numeral 28 (PSDP); FIG. 4, reference numerals 23 (disk), 28 (PSDP), 400 (Data Engine), 402 (DMA FIFO/driver), 404 (CPU interface), 406 (memory FIFO/driver), 407 (disk FIFO/driver), and 408 (disk interface); and FIG. 5, reference numerals 500 (filter logic), 502 (data parser), 504 (header storage), 506 (error checking), 508 (output tuple generator) and 510 (transaction ID processing) of Applicants' application as originally filed.

Claim 2

Claim 2 depends from Claim 1 and further requires that the use or lose decision value indicates a result from logic processing of fields read from the streaming data interface. Support for Claim 2 can be found at least on page 5, lines 25 to page 6, line 10 of Applicants' application as originally filed.

Claim 3

Claim 3 depends from Claim 1 and further requires that the use or lose decision value indicates a result from logic processing of fields read from the streaming data interface. Support for Claim 9 can be found at least on page 6, lines 11-27 of Applicants' application as originally filed.

Claim 4

Claim 4 depends from Claim 3 and further requires that TID processing and data engine logic execute in parallel. Support for Claim 4 can be found at least on page 2, lines 18-27 of Applicants' application as originally filed.

Claim 5

Claim 5 depends from Claim 1 and further requires that the output tuple is greater in length than an expected predetermined size, and the use or lose decision value is then used to set an overflow field in the output tuple. Support for Claim 5 can be found at least on page 8, lines 4-18 and page 33, lines 1-10 of Applicants' application as originally filed.

Claim 6

Claim 6 depends from Claim 5 and further requires means for appending an overflow filter bit to a tuple that indicates a transfer of a tuple that should be ignored when the use or lose decision value is not asserted when a buffer local to the programmable data streaming processor is full. Support for Claim 5 can be found at least on page 8, lines 4-18 of Applicants' application as originally filed.

Claim 8

Claim 8 depends from Claim 1 and further requires that the use or lose decision value is used to reset the output FIFO write pointer so any prior fields in the present tuple are discarded. Support for Claim 8 can be found at least on page 7, lines 21-28 of Applicants' application as originally filed.

Claim 9

Claim 9 depends from Claim 1 and further requires that the overflow filter bit is inserted in a length field appended to record fragments. Support from Claim 9 can be found at least on page 8, lines 4-18 of Applicants' application as originally filed.

Claim 10

Claim 10 depends from Claim 1 and further requires that an invalid field is appended to a tuple to indicate the results of TID processing. Support for Claim 10 can be found at least on page 30, lines 9-11 of Applicants' application as originally filed.

Claim 13

Claim 13 depends from Claim 1 and further requires a register reflecting the final PSDP status which is read by a CPU to identify whether any overflow or TID status bits are set in any of the tuples. Support for Claim 13 can be found at least on page 33, lines 1-6 of Applicants' application as originally filed.

Claim 14

Claim 14 depends from Claim 1 and further requires that the use or lose decision value represents DeMorgan's Law reduction of multiple instructions. Support for Claim 14 can be found at least on page 27, lines 19-25 of Applicants' application as originally filed.

VI. GROUND OF REJECTION TO BE REVIEWED ON APPEAL

A. Whether Claims 1-14 are properly rejected under 35 U.S.C. § 103(a) as being obvious over U.S. Patent 6,434,649 to Baker et al. (hereinafter, "Baker") in view of U.S. Patent Application Publication 2003/0126056 to Hausman et al. (hereinafter, "Hausman").

VII. ARGUMENT

A Final Office Action dated June 11, 2008 (hereinafter, the Final Office Action) rejected claims 1-14 under 35 U.S.C. § 103(a) as being obvious over Baker in view of Hausman. Applicants submitted a Reply After Final Rejection on July 7, 2008 and, in response, received an Advisory Action Before the Filing of an Appeal Brief dated July 28, 2008 (hereinafter, the Advisory Action).

By way of review, example embodiments of Applicants' claimed invention are directed to a Programmable Streaming Data Processor (PSDP) that performs primitive initial processing functions on a set of data, directly as that data is received from a streaming data source. The PSDP processes data from a streaming data source, such as a disk drive, prior to its being forwarded to a central processing unit (CPU) of a more general processor. The PSDP performs certain preliminary processing in order to reduce the computational load on the local CPU.

The PSDP may have processing logic known as a Data Engine that is capable of determining field boundaries in streaming data and processing fields to select one or more fields to be assembled into output tuples. An output tuple is comprised of the fields of the source from the disk that are to be selected for further processing by the CPU and PSDP generated fields. The Data Engine then determines whether the output tuple is to be selected for further processing and asserts a use or lose decision value according to that determination by including that value along with the fields selected to be the output tuple. Output tuples are then assembled from the selected fields and, if the use or lose decision value indicates that an output tuple is to be discarded, are prevented from being transferred for further processing.

Baker relates to a data processor, and more specifically to a data transfer arrangement mechanism employed to transfer data to various components within a data processor. One such data transfer arrangement is directed to processing computer graphics and graphics on a standalone gaming console.

Hausman relates to distribution of financial data. Users of terminals in a computer system select data from feeds provided over a network. The data may be used in data-requesting programs accessed at the terminals. Data identified for distribution to applications are mapped into forms as specified by the users.

1. **The Examiner Fails to Meet the Burden of Showing a Prima Facie Case of Obviousness**

Applicants have carefully considered the Examiner's comments and arguments in both the Final Office Action and the Advisory Action while preparing these Arguments. However, the Examiner has not set forth a *prima facie* case that the Applicants' claims are obvious. Baker and Hausman, individually or in combination, do not disclose or suggest all of the elements of the independent claim. Applicants also do not agree that one of ordinary skill in the art would be motivated to combine the references as suggested by the Examiner.

The Examiner bears the burden of establishing a prima facie case of obviousness. In order for the Examiner to support a prima facie case of obviousness, all claim limitations must be considered. "All words in a claim must be considered in judging the patentability of that claim against the prior art." Manual of Patent Examining Procedure 8 Ed. Rev. 7 (July 2008) (MPEP) § 2143.03, citing *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970).

The failure of an asserted combination to teach or suggest each and every feature of a claim remains fatal to an obviousness rejection under 35 U.S.C. § 103, despite any recent revision to the MPEP. Although MPEP § 2143.03 requires the "consideration" of every claim feature in an obviousness determination, to render Claim 1 unpatentable, the Examiner must do more than merely "consider" each and every feature for this claim. Instead, the asserted combination of the patents to Baker and Hausman must also teach or suggest each and every claim feature. See *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). To establish a prima facie case of obviousness under 35 U.S.C. § 103(a), the Examiner must demonstrate that each and every claim limitation is taught or suggested by the cited references, and that a combination of such elements would be obvious to one of ordinary skill in the art. MPEP § 2143.

As the Board of Patent Appeals and Interferences has recently confirmed, a proper obviousness determination requires that an Examiner make "a searching comparison of the claimed invention – including all its limitations – with the teaching of the prior art." See *In re Wada and Murphy*, Appeal 2007-3733 (BPAI 2008), citing *In re Ochiai*, 71 F.3d 1565, 1572 (Fed. Cir. 1995) (emphasis in original). Further, the necessary presence of all claim features is axiomatic, because the Supreme Court has long held that "obviousness is a question of law based on underlying factual inquiries, including ... ascertaining the differences between the claimed

invention and the prior art.” *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966) (emphasis added). Indeed, Applicants submit that this is why MPEP § 904 instructs Examiners to conduct an art search that covers “the invention as described and claimed.” (emphasis added). Lastly, MPEP § 2143 buttresses the conclusion that obviousness requires at least a suggestion of all of the features of a claim by citing from the Supreme Court in *KSR Int’l v. Teleflex Inc.* that “there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.” 127 S. Ct. 1727, 1741 (2007) (quoting *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006)).

In sum, it remains well-settled law that obviousness requires at least a suggestion of all of the features in a claim. See *In re Wada and Murphy*, citing *CFMT, Inc. v. Yieldup Intern. Corp.*, 349 F.3d 1333, 1342 (Fed. Cir. 2003) and *In re Royka*, 490 F.2d 981, 985 (CCPA 1974)). Further, if an independent claim is nonobvious under 35 U.S.C. § 103, then any claim depending therefrom is nonobvious. MPEP § 2143.03, citing *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988). Because multiple elements of independent Claim 1 are not found in the references, and Claims 2-14 depend from Claim 1, the rejection of all Claims 1-14 should be withdrawn.

A. Neither Baker nor Hausman performs primitive initial processing functions on data from a streaming data source; Baker and Hausman simply route data within the same data processor and to clients requesting it, respectively

The Examiner argues that Baker’s multimedia processor 100 is the claimed PSDP because it comprises the same components and is a data processor for streaming. However, Baker’s processor 100 does not perform primitive initial processing functions directly on data received from a streaming data interface as required by Applicants’ claims.

In the example embodiment of the present invention illustrated in FIG. 1 of Applicants’ published application, a distributed data processing system uses multiple processing unit groups and a processor (PSDP 28) at each JPU 22 performs initial primitive functions on data from a streaming data source (disk 23) before that data is further handled by a more general purpose job processor (μP 26). As can be seen in FIG. 1 of the Applicants’ published application, each JPU 22 has one or more mass storage devices, such as a disk 23, attached from which the JPU 22 may read streaming data. Therefore, the PSDP 28 operates on the raw data from its associated disk 23 and forwards it to another processor 26. Example processing functions that Applicants’

claimed PSDP can perform include selecting fields to be assembled into output tuples and, as described in paragraph [0073] of Applicants' published application, tagging records as they are processed to indicate that such records are to be ignored for further processing or to indicate certain attributes of such records (*e.g.*, they are to be handled differently in a transaction from other records) and other non-filter like processes, such as compression/decompression, encryption/decryption, simple join operations, and the like.

However, with regard to Baker, the data transfer switch 112 is simply a routing mechanism for switching connections within a processor and does not perform any primitive initial processing functions on data. As described in its abstract, Baker teaches "a data transfer switch for performing data transfer operations" (emphasis added). All of the components of the Baker's multimedia processor 100 are coupled to the data transfer switch 112: data cache 108, instruction cache 110, fixed function cache 106, memory controller 124, data streamer 122, PCI/AGP interface 130, and programmable input/output controller 126. Nowhere in Baker does it describe the multipurpose processor 100 as performing any primitive initial processing functions on the data that passes through the data transfer switch 112, as does Applicants' claimed PSDP.

B. Hausman neither selects fields to be assembled into output tuples nor assembles fields into tuples; Hausman merely passes complete records to clients requesting them

Hausman also does not select fields to be assembled into output tuples as in Claim 1. As described in paragraph [0108] of Applicants' published application, the term "tuple" is used by Applicants for the purpose of differentiating "raw" disk 23 record formats from PSDP 28 output record formats. For example, paragraph [0019] of Applicants' published application teaches that an output tuple "is comprised of the fields of the source record from the disk that are to be selected for further processing by the CPU and PSDP generated fields."

The Examiner has likened application programming interface (API) 104 of Hausman to Applicants' claimed data engine. However, as described in paragraph [0045] of Hausman, the API 104 "selects...data records requested by users served by the API" (emphasis added). Thus, Hausman describes wholesale selection by API 104 of entire requested records (*i.e.*, raw data), and not selection of particular fields of those records (*i.e.*, an output tuple comprised of the fields

of the source record from the disk that are to be selected for further processing by the CPU and PSDP generated fields). Moreover, by the Examiner's own admission, "according to [0045], only the records that are requested by the users are sent to that particular user [JPU] for processing" (emphasis added). Hausman has no teaching, mention or even a suggestion of tuples as required by Applicants' claims.

During examination, the claims must be interpreted as broadly as their terms reasonably allow. *In re Am. Acad. of Sci. Tech Center*, 367 F.3d 1359, 1369 (Fed. Cir. 2004). However, when the Specification of a patent application explicitly states the meaning that a term in the claim is intended to have, the claim is examined using that meaning, in order to achieve a complete exploration of the applicant's invention and its relation to the prior art. *In re Zletz*, F.2d 319, 321-22 (Fed. Cir. 1989).

Therefore, the Examiner's interpretation of the claim term "tuple" is unjustified. Given the explicit intended meaning of the term recited in paragraphs [0019] and [0108] of Applicants' published application, equating both a tuple and a record to a "row" of data is inconsistent.

It then stands that Hausman also does not assemble fields into an output tuple as required by Applicants' Claim 1. As emphasized above, an output tuple "is comprised of the fields...that are to be selected for further processing by the CPU and PSDP generated fields" (paragraph [0019] of Applicants' published application). These selected fields, plus those assembled by the claimed tuple generator, do not equate to a row of raw data from the data source as asserted by the Examiner. Rather, they comprise a new data form (*i.e.*, a tuple) generated by the tuple generator of the PSDP. A row of data, as in Hausman, equates to a database record but does not equate to a tuple. Therefore, Hausman also fails to teach assembling fields into output tuples.

C. *Housman does not determine field boundaries in output data; Hausman reviews fields already delineated at the streaming data source*

Hausman does not determine field boundaries in output data from a streaming data source as required by Applicants' Claim 1. As described in paragraph [0015] of Applicants' published application:

...the PSDP can parse non-field-delineated, streaming data from the [streaming data source] into block header fields, record header fields, and record data fields...In other words, the PSDP can be programmed to understand the record and field structure of the data which the analysis software running on the CPU of the JPU wishes to analyze. Therefore, the PSDP can further process data in the specific format of the database application (emphasis added).

The Examiner has likened Applicants' claimed data engine to the API in Hausman. For determination of field boundaries, the Examiner looks to paragraph [0039] of Hausman which teaches the insertion of delimiters between records and between elements within the records. However, paragraph [0039] makes it clear that delimiters are inserted at the data stream source 101, and not at the API 103. In fact, the data stream source 101 in Hausman is well upstream of the API 104 at the client server 105. Thus, Hausman's API does not determine field boundaries in data from the streaming interface. Rather, the data, received by Hausman's API from the streaming interface, is already field-delineated. This interpretation of Hausman is actually supported by the Examiner's admission at the bottom of page 2 of the Advisory Action that "the output data [as received by API 104 from queue 108] includes delimiters between records and between elements within the records."

As also described in paragraphs [0043] and [0045] of Hausman:

Data from source 101 is received through queue 108

and

API 104 reviews data records included within the stream [received from source 101 through queue 108]...and selects...data records requested by users served by the API (emphasis added).

Hausman's API therefore is limited to reviewing data records already delineated into fields.

D. Hausman neither asserts a use or lose decision nor prevents the output tuple from being transferred according to a use or lose decision value; Hausman simply routes or filters records

Hausman also does not assert a use or lose decision after determining that an output tuple is to be selected for further processing as required by Applicants' Claim 1. Specifically, as described in paragraph [0121] with reference of FIG. 5 of Applicants' published application, a use/lose decision bit (value) 520 is part of the output that is fed to the tuple generator 508. Therefore, the use/lose decision bit 520 is sent (*i.e.*, asserted) along with the fields selected to be the output tuple. The tuple generator discards or presents the tuple, depending on the use or lose decision value passed to it with the tuple.

Hausman makes no mention of anything corresponding to a use or lose decision being asserted; Hausman only filters and routes data according to the data and does not pass another generated value with the data.

Further, Hausman does not prevent a formed output tuple from being transferred if the use or lose decision value indicates that such output tuple is to be discarded. Referring again to Applicants' published application, if the use/lose decision value 520 indicates that the tuple is to be discarded, the tuple generator 508 then prevents that particular tuple from being forwarded to the memory of the JPU (FIG. 5 as described in paragraph [0121]).

The Examiner argues that the concepts in Hausman of deleting elements from records and not sending particular records to particular users is considered to represent the claimed "use or lose" decision. As described in paragraph [0057] of Hausman, the API 104 simply maps records of the data stream to individual users 106. The Examiner does not explicitly state where Hausman teaches a use or lose decision that results in "not sending particular records." Applicants note that this is, at best, further described in paragraph [0045] with regard to "mapping" the records. However, mapping records does not equate to preventing particular records from being sent at all. Rather, although a particular user only receives requested records, any remaining records in the mapping (*e.g.*, those requested by other users) are indeed sent, even though not sent to the particular user, but to other users. Such mapping does not prevent data from being transferred but rather actually transfers the data to various users (*e.g.*, all those requesting it).

Further, merely filtering records according to user criteria, as in Hausman, does not teach the claimed generating of a use or lose decision value as output to the tuple generator and asserted as claimed. Although paragraph [0065] of Hausman teaches filtering records identified for specific users that do not match user criteria, there is no teaching of asserting a use or lose decision value. Contrary to the Examiner's assertion that filtering is analogous to the use or lose decision value, filtering only determines to which user data should be sent, not if that data should be sent at all, and certainly not the same as storing the result of the decision as a data value to be carried along with the tuple. Thus, the claimed use or lose decision value is also not taught by Hausman.

E. Hausman does not contain logic; Hausman selects records based in software

Finally, Hausman does not contain logic arranged to determine whether an output tuple is to be selected for further processing as in Claim 1. As described in paragraph [0072] of Applicants' published application, the data engine, as a component of the PSDP, is programmable hardware ("a finite state machine" as further described in paragraph [0083]) disposed directly in the read path of a streaming data source, such as a disk drive. Applicants' data engine thus allows the PSDP to be programmed to operate on data as it is received from the disk before it is stored in the JPU's memory and, in the process, discard data that the more general purpose CPU would otherwise have to analyze and discard in the absence of the data engine.

In contrast, the API 104, described in paragraph [0043] of Hausman as "programming causing client system 192 to select, deliver, and optionally map data records received or retrieved in a stream from source 101," is not the claimed data engine (emphasis added). As understood in the art, an API is a software package, typically including a set of routines, protocols, and tools, for building software applications on any general purpose processor. Hausman makes no mention of logic as in Claim 1. Therefore, Hausman also fails to teach a data engine containing logic to determine whether an output tuple is to be selected for further processing by additional JPUs.

2. **It would not be obvious to combine the teachings of Baker and Hausman**

According to the reasoning which is to be applied per the MPEP § 2141 in making an obviousness rejection:

The key to supporting any rejection under 35 U.S.C. 103 is the clear articulation of the reason(s) why the claimed invention would have been obvious. The Supreme Court in KSR noted that the analysis supporting a rejection under 35 U.S.C. 103 should be made explicit. The Court quoting *In re Kahn*, 441 F.3d 977, 988, 78 USPQ2d 1329, 1336 (Fed. Cir. 2006), stated that "[R]ejections on obviousness cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." KSR, 550 U.S. at ___, 82 USPQ2d at 1396.

The Examiner's reasoning, as admitted by the Examiner in the Advisory Action, fits best under the rationale discussed under part (G) of MPEP § 2143, "Some Teaching, Suggestion, or Motivation in the Prior Art That Would Have Led One of Ordinary Skill To Modify the Prior Art Reference or To Combine Prior Art Reference Teachings To Arrive at the Claimed Invention":

To reject a claim based on this rationale, Office personnel must resolve the Graham factual inquiries. Then, Office personnel must articulate the following:

(1) a finding that there was some teaching, suggestion, or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings;

(2) a finding that there was reasonable expectation of success; and

(3) whatever additional findings based on the Graham factual inquiries may be necessary, in view of the facts of the case under consideration, to explain a conclusion of obviousness.

The rationale to support a conclusion that the claim would have been obvious is that "a person of ordinary skill in the art would have been motivated to combine the prior art to achieve the claimed invention and that there would have been a reasonable expectation of success." *DyStar Textilfarben GmbH & Co. Deutschland KG v. C.H. Patrick Co.*, 464 F.3d 1356, 1360, 80 USPQ2d 1641, 1645 (Fed. Cir. 2006). If any of these findings cannot be made, then this rationale cannot be used to support a conclusion that the claim would have been obvious to one of ordinary skill in the art.

The Final Office Action states on page 5 that it would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine Hausman's method of filtering data as a subcomponent to Baker's data streamer because "it is well known to one of ordinary skill that filtering provides customized distribution of data and also decreases...the amount of information sent across the network." In the Advisory Action on page 3, the Examiner further states, as motivation to combine the references:

...Hausman provides the feature of [being] able to filter data. It is well known to one of ordinary skill in the art that filtering data decreases the amount of data. In this case, the data is being sent across a network and therefore reducing the amount of information of data would reduce for example, the required bandwidth. The results of decreasing the amount of data sent across a network are well known in the art.

A. There is no teaching, suggestion or motivation to so combine the references

While the Examiner's assertion is true with respect to the teachings of Hausman, which sends data over a network from the client server 105 to application terminals 106, one of ordinary skill in the art would not be so motivated to combine Baker and with Hausman. Baker relates to a data processor, and more specifically to a data transfer arrangement mechanism employed to transfer data to various components within that very same data processor. Although Baker does discuss bandwidth requirements between particular input/output (I/O) devices, this is in context to supporting data transfers between endpoints that exhibit mismatched or varying bandwidth requirements within a data processor, and has nothing to do with reducing bandwidth requirements over a network as in Hausman. Therefore, because there is no network in Baker in which bandwidth requirements may be reduced, one of ordinary skill in the art would not look to combine the Baker and Hausman references.

B. The proposed modification of Baker by Hausman would render Baker unsatisfactory for its intended purpose

Further, the Final Office Action is further deficient for failing to indicate how such a combination would have had a reasonable expectation of success. Quite to the contrary, one of ordinary skill in the art would not be motivated to combine Baker and Housman because the combination would render Baker unsatisfactory for its intended purpose. According to MPEP § 2143.01(I)(V):

If proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification (emphasis added).

In this case, by combining the Baker and Hausman references, as suggested by the Examiner, the Baker system fails for its intended purpose.

Specifically, Baker's data streamer is employed for predetermined data movements within a multimedia processor (col. 5, lines 58-59). That is, Baker's data streamer specifically supports data transfer between memory or I/O devices within the same processor. Thus, one skilled in the art would not be motivated to combine the network data bandwidth reduction of Hausman with the data streamer of Baker because Baker, as modified by Hausman (*i.e.*, filtering out data), would fail to deliver all data from the source, such as the memory. Baker would then fail to properly transfer data to the various internal components of the data processor.

Accordingly, the proposed modification of Hausman renders Baker unsatisfactory for its intended purpose. That is, Baker's providing of buffered data movements within a multimedia processor cannot be executed by using the reduction in transmitted data provided by the network data filtering taught in Hausman.

C. The proposed combination would not result in the claimed invention

Further, the Examiner's reasoning (*i.e.*, reducing required bandwidth) is not relevant to Applicants' stated purpose (*i.e.*, reducing the computational load on the local CPU), and such a combination would not result in the claimed invention. Even if Baker and Hausman were combined in the manner described by the Examiner, such a combination would not "reduce the computational load on the local CPU" as described in paragraph [0009] of Applicants' published application. Rather, reducing the amount of data sent across a network reduces the required network bandwidth. This is irrelevant to Applicants' invention because, even if data were streamed over a network in example embodiments of the present invention, the data has already been streamed from the disk 23 to the PSDP 28 before it reaches the data engine where primitive initial processing functions are performed on the data. Therefore, one of ordinary skill in the art would never combine Hausman with Baker to determine how to "reduce the computational load on the local CPU."

3. Claim Rejections

Claim 1

It is thus respectfully submitted that the Examiner neither has shown where in the prior art references each of the elements of Claim 1 is found nor presented a valid motivation for combining the references. The combination of Baker and Hausman does not overcome any of the deficiencies of Baker. Because the Final Office Action is so deficient, and fails to clearly articulate the reason(s) why the claimed invention would have been obvious, the Final Office Action fails to set out a prima facie case of obviousness. Further, all other Claims 2-14 depend from independent Claim 1 and contain all the elements of the base claim. At least for this reason these claims are thus nonobvious as well.

Claim 2

With regard to the Examiner's rejection of Claim 2, the combination of Baker and Hausman does not overcome any of the deficiencies of Baker and fails to disclose results from logic processing of fields read from the streaming data interface being indicated in the use or lose decision value, let alone a use or lose decision value or logic processing. The Examiner relies on Baker column 17, line 52 to column 18, line 12. However, this portion of Baker teaches the use of a transfer engine to implement a pipeline control logic to handle simultaneous data transfers between different components of Baker's multimedia processor 100 and has nothing to do with the use or lose value indicating a result from logic processing of fields read from the streaming data interface.

Further, in making the rejection of Claim 1, the Examiner admitted that Baker "fails to explicitly teach the further limitation[] of the data engine..." As recited in Claim 1, the data engine asserts a use or lose decision value according to the determination of whether an output tuple is to be selected for further processing. It then follows that Baker cannot teach the use or lose value indicating a result from logic processing of fields read from the streaming data interface, as in Claim 2, if Baker does not teach a use or lose value.

Claims 3 and 4

With regard to the Examiner's rejections of Claims 3 and 4, the combination of Baker and Hausman does not overcome any of the deficiencies of Baker and fails to disclose either a Transaction Identifier (TID) or TID processing, let alone TID processing and data engine logic being executed in parallel. The Examiner argues that the routing and address tags disclosed in paragraphs [0063] and [0064] are transaction identifiers. However, as understood by one of ordinary skill in the art, a TID is a unique identifier assigned to each database transaction, used to identify all actions associated with that transaction. Association of individual records with routing or address tags has nothing to do with identifying database transactions. They simply identify to the API to which client each record should be routed.

Claims 5 and 6

With regard to the Examiner's rejections of Claims 5 and 6, the combination of Baker and Hausman does not overcome any of the deficiencies of Baker and fails to disclose either using the use or lose decision value to set an overflow field in the output tuple when the output tuple is greater in length than an expected predetermined size, or means for appending an overflow filter bit to a tuple that indicates a transfer of a tuple should be ignored when the use or lose decision value is not asserted when a buffer local to the programmable data streaming processor is full, let alone tuples or use or lose decision values. The value of the valid bit used in Baker indicates whether the specific byte is valid or not. Baker does not, however, not assert a use or lose decision value when a buffer local to the programmable data streaming processor is full and make no use of an overflow filter bit to a tuple that indicates a transfer of a tuple that should be ignored.

Claim 8

With regard to the Examiner's rejection of Claim 8, the combination of Baker and Hausman does not overcome any of the deficiencies of Baker and fails to disclose the use or lose decision value being used to reset the output FIFO write pointer so any prior fields in the present tuple are discarded. The Examiner relies on column 12, lines 18-34 of Baker. However, this portion of Baker discusses the initiation phase in a write transaction between two units of

Baker's multimedia processor 100 and has nothing to do with the use or lose value being used to reset the output FIFO write pointer so any prior fields in the present tuple are discarded.

As described in paragraphs [0020]-[0021] of Applicants' published application, as data streams out of the filter, an output tuple is formed in a FIFO memory in a way that permits aborting the tuple if the filter logic determines that the particular tuple should not be passed on to the CPU. Specifically, the memory FIFO has two write pointers, an "active" write pointer and a "visible" write pointer. The visible pointer maintains a position indicating a boundary of the last accepted tuple. Meanwhile, the active write pointer moves along the memory FIFO from the boundary, as words of the next possible tuple become available. As the PSDP logic determines that a tuple is not to be used, such as a result of the filter or TID processing described above, the memory FIFO's active write pointer resets by moving back to the visible write pointer location. This has the effect of ignoring the intervening fields of the unwanted tuple and allowing them to be overwritten. If the PSDP logic makes this determination while the active pointer is still pointed to a field within the unwanted tuple, the active pointer will simply reset to the visible pointer location until the last field within that unwanted tuple has been overwritten. If, on the other hand, the PSDP logic determines that a tuple is to be used, the visible pointer moves to the active pointer position, having the effect of keeping all intervening fields of the tuple that should be kept. There is no such teaching in Baker of write pointers or resetting write pointers according to the use or lose decision value.

Claims 9 and 10

With regard to the Examiner's rejection of Claims 9 and 10, the combination of Baker and Hausman does not overcome any of the deficiencies of Baker and fails to disclose either an overflow filter bit or tuples, let alone an invalid field appended to a tuple or such an overflow filter bit inserted in a length field appended to record fragments. In making these rejections in the Final Office Action, the Examiner relied on column 34, lines 56-62 and column 12, line 62 to column 13, line 16, respectively. Following Applicants' arguments in the Reply After Final Rejection, the Examiner now generally asserts that "Hausman discloses parsing the filter which is considered to represent this step." However, nowhere does Hausman teach an overflow filter bit inserted in a length field appended to record fragments or an invalid field appended to a tuple to indicate the results of transaction ID processing.

Claim 13

With regard to the Examiner's rejection of Claim 13, the combination of Baker and Hausman does not overcome any of the deficiencies of Baker and fails to disclose a register reflecting the final PSDP status which is read by a CPU to identify whether any overflow or TID status bits are set in any of the tuples. As discussed in paragraph [0129] of Applicants' published application, when the data engine sets an invalid bit or an overrun bit in any given tuple, it can also set a flag in a read register which can be used as a flag to summarize the status bits of all tuples in a related group. This read register can reflect the final PSDP status to identify whether any overflow or TID status bits were set in any of a group of tuples, further relieving the CPU from having to read the status bits of all tuples to determine if any have caused an overflow condition. There is no such teaching in Baker as cited by the Examiner.

Claim 14

With regard to the Examiner's rejection of Claim 14, the combination of Baker and Hausman does not overcome any of the deficiencies of Baker and fails to disclose a representation of DeMorgan's Law reduction of multiple instructions. Baker, column 5, lines 25-34 discusses how the compiler in a Very Long Instruction Word (VLIW) architecture must assemble many primitive operations into a single "instruction word." This does not teach the rules of formal logic relating pairs of dual logical operators in a systematic manner expressed in terms of negation.

Claims 7, 11 and 12

With regard to the Examiner's rejections of Claims 7, 11 and 12, the combination of Baker and Hausman does not overcome any of the deficiencies of Baker.

4. **Conclusion**

In view of the foregoing arguments, Applicants respectfully submit that all claims remaining in the application are in condition for allowance. Reversal of the rejections is requested so the application may pass to issue.

Respectfully submitted,

HAMILTON, BROOK, SMITH & REYNOLDS, P.C.

By


Gerald P. Kazanjian

Registration No.: 61,699

Telephone: (978) 341-0036

Facsimile: (978) 341-0136

Concord, MA 01742-9133

Dated: 10/1/2008

VIII. CLAIMS APPENDIX

1. A Programmable Streaming Data Processor (PSDP), arranged to perform primitive initial processing functions directly on a set of data comprising:
 - a streaming data interface, arranged to receive data from a streaming data source;
 - a streaming interface First In First Out (FIFO), arranged to temporarily store streaming data from the streaming data interface;
 - a data engine, arranged to receive output data from the streaming interface FIFO, determine field boundaries therein, and process fields to select one or more fields to be assembled into output tuples, the data engine also containing logic arranged to determine whether an output tuple is to be selected for further processing by additional Job Processing Units (JPUs) and to assert a use or lose decision value according to that determination;
 - a tuple generator, arranged to assemble fields into the output tuple and, if the use or lose decision value indicates that such output tuple is to be discarded, to prevent the output tuple from being transferred for further processing by the JPU; and
 - an output FIFO device, arranged to temporarily store tuples prior to conditionally forwarding them to the JPU.
2. An apparatus as in claim 1 wherein the use or lose decision value indicates a result from logic processing of fields read from the streaming data interface.
3. An apparatus as in claim 1 wherein the use or lose decision value indicates a result from Transaction Identifier (TID) processing.

4. An apparatus as in claim 3 wherein the TID processing and data engine logic execute in parallel.
5. An apparatus as in claim 1 wherein the output tuple is greater in length than an expected predetermined size, and the use or lose decision value is then used to set an overflow field in the output tuple.
6. An apparatus as in claim 5 wherein the use or lose decision value is not asserted when a buffer local to the programmable data streaming processor is full; and
means for appending an overflow filter bit to a tuple that indicates a transfer of a tuple that should be ignored.
7. An apparatus as in claim 1 additionally comprising:
a Direct Memory Access (DMA) interface, coupled to the output FIFO, to provide direct access to a memory in the JPU.
8. An apparatus as in claim 1 wherein the use or lose decision value is used to reset the output FIFO write pointer so any prior fields in the present tuple are discarded.
9. An apparatus as in claim 1 wherein the overflow filter bit is inserted in a length field appended to record fragments.
10. An apparatus as in claim 1 wherein an invalid field is appended to a tuple to indicate the results of TID processing.

11. An apparatus as in claim 10 wherein the invalid field indicates that the TID mode marks return tuple.
12. An apparatus as in claim 10 wherein the invalid field indicates that the tuple should not have been returned but the output FIFO overflowed.
13. An apparatus as in claim 1 further comprising:
a register reflecting the final PSDP status which is read by a Central Processing Unit (CPU) to identify whether any overflow or TID status bits are set in any of the tuples.
14. An apparatus as in claim 1 wherein the use or lose decision value represents DeMorgan's Law reduction of multiple instructions.

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None. However, as noted above in Section II, Related Appeals and Interferences, a related application, U.S. Patent Application No. 10/665,726, filed September 18, 2003, has received a Final Office Action mailed from the Office on July 25, 2008. Applicants may file an appeal in that related application as well.